

## 9. Abend

Ziel :

- Wir wissen ab heute Abend wie wir eindeutige Namen in C++ erzeugen, indem wir Namensbereiche definieren und anwenden und verwenden mehrere Dateien in einem Projekt.

### Namespaces, Namensräume

Wir erzeugen zuerst eine eigene Bibliothek und nennen sie Mathe1 :

Erzeuge im Visual-Studio zuerst eine Header-Datei (mit Menu-File->neu...) und nenne sie Mathe1.h. In dieser Bibliothek haben wir die Funktion Berechnen, die aus einem int etwas berechnet und das Ergebnis als Rückgabewert hat. In der Header-datei haben wir aber nur den Funktionsprototypen !

```
int Berechne(int a);
```

Die Definition der Funktion packen wir in eine weitere neue Datei, aber diesmal eine Quellcodedatei mit der Endung .cpp. Sie heisst sonst aber gleich wie die zugehörige Header-Datei, also hier Mathe1.cpp

```
// Der Compiler muss wissen
// worum es geht, d.h. er braucht
// den Funktionsprototypen
#include "Mathe1.h"

int Berechne(int a)
{
    return a*a;
}
```

Erzeuge nun eine neue Bibliothek mit dem Namen Mathe2, also Mathe2.h und Mathe2.cpp. In dieser Bibliothek soll ebenso eine Funktion Berechne sein, die also genau gleich aussieht wie die erste Berechne-Funktion.

Wenn wir nun als letztes ein main.cpp mit folgender main-Funktion erstellen und die beiden Funktionen brauchen wollen haben wir ein Problem

```
#include "Mathe1.h"
#include "Mathe2.h"

int main()
{
    int a = 4;

    a = Berechne(a);

    int b = 8;

    b = Berechne(b);

    return 0;
}
```

Der Compiler kann jetzt nicht mehr unterscheiden welche Funktion gemeint ist ! Angenommen die beiden Bibliotheken stammen nicht von uns sondern von zwei verschiedenen Herstellern, wir können diese also nicht ändern und haben nun ein Problem, da mehrerer Funktionen den gleichen Namen haben. Durch Definition eines Namensraumes können wir das Problem lösen :

#### **Mathe1.h**

```
namespace ACME
{
    int Berechne(int a);
};
```

#### **Mathe1.cpp**

```
#include "Mathe1.h"

int ACME::Berechne(int a)
{
    return a*a;
}
```

Der Namensraum heisst hier also ACME. Mit ACME:: (die zwei Doppelpunkte heissen Sichtbarkeitsoperator) gelangen wir in den Namensraum ACME. In Mathe2 verwenden wir eine anderen Namen :

**Mathe2.h**

```
namespace Chiqita
{

int Berechne(int a);

};
```

**Mathe2.cpp**

```
#include "Mathe2.h"

using namespace Chiqita;

int Berechne(int a)
{
    return a*a*a;
}
```

Hier jetzt das main mit Aufrufen für die verschiedenen Varianten :

**main.cpp**

```
#include "Mathe1.h"
#include "Mathe2.h"

int main()
{
    int a = 4;

    a = ACME::Berechne(a);

    int b = 8;

    using namespace Chiqita;

    b = Berechne(b);

    return 0;
}
```

Diese zweite Schreibweise kennen wir doch bereits ! Wenn wir nämlich eine Variable vom Datentyp string oder die cin/cout-Objekte verwenden.

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
...
}
```

Der neue C++ Standard verwendet gewisse Namen, die bereits früher verwendet wurden noch einmal. Bevor nämlich der C++ Standard verabschiedet wurde, haben viele Compiler-Hersteller bereits eigene Erweiterungen vorgenommen ! So gibt es zum Beispiel auch eine Bibliothek <iostream.h> (beachte das .h als Endung). Auch in jener Bibliothek gibt es ein cin und ein cout !

Verwendet aber trotzdem den neuen C++ Standard, das heisst <iostream> mit using namespace std;